

OPTIMIZACIÓN - 1ER CUATRIMESTRE 2017

Trabajo Práctico 3 - Heurísticas

Fecha de entrega: Viernes 23 de junio.

1 Objetivo del Trabajo

El objetivo del trabajo es realizar un estudio exhaustivo de algunos algoritmos heurísticos. El trabajo requerirá de la confección de un informe final con los resultados obtenidos. Las funciones y rutinas no necesitan ser diseñadas para el usuario, aunque sí deberán mantener los principios de programación que fueron delineados en las sucesivas clases y el código deberá estar comentado.

Se deberán implementar *todos* los siguientes algoritmos heurísticos:

A1 Descenso aleatorio.

- Descenso más rápido.
- Búsqueda local.
- Búsqueda local iterada.

A2 Algoritmos genéticos.

A3 Recocido simulado.

A4 Algún algoritmo extra para cada grupo.

Los problemas propuestos a minimizar son:

P1 Minimizar la función $f(x, y) = (x - 0.5)^2 + (y - 0.75)^2$ en el cuadrado $[0, 1] \times [0, 1]$.

P2 Minimizar la función de Rosenbrock $f(\bar{x}) = \sum_{i=1}^{n-1} (1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2$ donde $x_i \in [-2.048, 2.048]$. Notar que se alcanza el mínimo en el vector $(1, 1, \dots, 1)$.

P3 En el archivo `testFunctions.zip` hay un gran conjunto de funciones para implementar los algoritmos de minimización. El mismo está explicado aquí.

También se pueden encontrar funciones aquí.

Intentar que si diferentes métodos usan iguales parámetros estén codificados de la misma manera.

Informe:

El informe deberá incluir tiempos de corrida de los algoritmos, distancia al mínimo (mejor aproximación conocida), parámetros que se usaron para las corridas. Un breve análisis comparativo de los métodos para las funciones estudiadas.

Incluir un gráfico del valor objetivo en función de cada iteración de los distintos algoritmos propuestos y compararlos.

2 Descenso aleatorio

Se basa simplemente en ir proponiendo en cada iteración n puntos factibles al azar, compararlos con el mejor hasta el momento y quedarse con el mejor. Considerar también la versión de descenso aleatorio con *búsqueda local*, es decir, en cada paso buscar n puntos al azar pero en un entorno del mejor hasta el momento. Por último, considerar el método de búsqueda local iterada, que se trata una mezcla de ambos métodos: se hace búsqueda local con reinicios al azar.

3 Recocido simulado

Es un método que tuvo inicios a mediados de los 80's. Es una variación de un método de descenso, como los descritos en la Sección 2, ya que se continuará escogiendo un punto donde decrezca el valor objetivo. La novedad es que dado un punto S , si se consigue un punto R donde no decrece el valor objetivo $f(S)$, se acepta igualmente el nuevo valor R con probabilidad

$$P(t, R, S) = e^{\frac{f(R)-f(S)}{t}},$$

donde $t \geq 0$. Notar que el exponente es negativo y que cuanto más alejado esté $f(R)$ de $f(S)$ más cerca de 0 estará la probabilidad de aceptar el valor nuevo R . El parámetro t , al que llamaremos *temperatura*, es a definir por el usuario; si t es muy grande, la probabilidad de aceptar el valor R será de casi 1, si t es muy chico se aceptarán pocos R que empeoren el valor objetivo obtenido hasta el momento. La idea es comenzar con t bien grande y que en las primeras iteraciones se acepten *casi* todos los puntos factibles y así hacer un paseo al azar por la región, luego ir disminuyendo la temperatura así de a poco se va pareciendo a un método de descenso como los descritos en la Sección 2.

4 Algoritmo libre

Podrán escoger algún método de los dados en la teórica, como el de enjambre por ejemplo, o bien elegir uno fuera de los vistos en clase.

References

- [1] Sean Luke, *Essentials of Metaheuristics*, Lecture Notes, 2nd edition (2015).
<https://cs.gmu.edu/~sean/book/metaheuristics/>